Covid-19 Twitter Sentiment Analysis Using Machine Learning

by

Muhammad Ali Shaikh

B.E., Mehran University of Engineering and Technology, Pakistan, 2011

A Report Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

Covid-19 Twitter Sentiment Analysis Using Machine Learning

by

Muhammad Ali Shaikh

B.E., Mehran University of Engineering and Technology, Pakistan, 2011

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member

(Department of Electrical and Computer Engineering)

ABSTRACT

The Internet is widely used by almost everyone. People are choosing social media applications to voice their thoughts and concerns given the daily growth of these platforms. They can be used to collect and analyze public sentiments for purchasing products, diseases, political debates, and socio economic developments. Businesses, governments, and individuals can benefit from analyzing these sentiments. Twitter is a massive, rapidly growing platform where users share their opinions on politics, sports, products, and other topics. Therefore, Twitter tweets are very useful for determining public sentiments.

Sentiment analysis is a method of determining if text indicates a negative, positive or neutral emotion. This report presents automated sentiment analysis of Covid-19 tweets using Machine Learning (ML). The ML classifiers used are Naive Bayes (NB), Simple Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF). The classifiers were implemented in Jupyter notebook using the Python programing language. Accuracy, F-measure, recall, precision, and execution time are considered as performance metrics. The results obtained indicate that the DT classifier is the best in terms of these metrics.

Contents

Table of Contents

List of Figures

List of Table

Glossary

AI …….. Artificial Intelligence

ML …… Machine Learning

DL …… Deep Learning

NB …… Naive Bayes

SVM …… Simple Vector Machine

NLTK ……..... Natural Language Tool Kit

DT …… Decision Tree

RF …… Random Forest

NLP …… Natural Language Processing

IDE ……. Integrated Development Environment

SMOTE …… Synthetic Minority Oversampling Technique

TFIDF ……Term Frequency Inverse Document Frequency

CNN ……. Convolutional Neural Network

LSTM …… Long Short Term Memory

GUI …… Graphical User Interface

ACKNOWLEDGMENT

I would like to thank:

Almighty Allah for his countless blessings on me and on all human beings.

My Parents, for their unconditional love, prayers, patience, and support.

Dr. T. Aaron Gulliver, for support, mentoring, guidance, teaching, and help.

My Teachers, religious and academic, for making me a good person.

My Siblings, for their motivation, love and support.

My Friends, for technical support, advice, and encouragement to pursue graduate studies at UVic, particularly Muhammad Ismail Mangrio, Atique Ahmed, and Salahuddin Jokhio.

My Colleagues, for their guidance and motivation.

"*Success is a ladder that cannot be climbed with your hands in your pockets.*"
Unknown

DEDICATION

This report is dedicated to my loving parents for their prayers, support, sacrifices and trust, especially my father who passed away while writing this report.

Chapter 1

Introduction

On January 30, 2020, the World Health Organisation (WHO) designated Coronavirus or Covid-19 a pandemic, and since then has been working tirelessly to contain it. Covid-19 is a unique viral disease named after the year it was first detected [1]. This disease has spread to many countries and the fight to stop it is being conducted around the world. Social media companies such as Facebook, Twitter, Instagram, and Reddit have been working actively to validate Covid-19 information on their platforms. This is because of the presence of misinformation which is an attempt to deceive or confuse people with misleading information.

Twitter is a social media networking site. Users tweet to a network of contacts from a device on this social media platform. Twitter tweets can be used by governments, businesses, and people to measure public opinion or sentiments regarding a topic, product, people, or event. The number of tweets created each day on Twitter is enormous, so it is important to automate sentiment analysis of tweets to make the task of evaluating public opinion tractable.

Sentiment analysis refers to the systematic recognition, extraction, evaluation, and examination of the emotional state of the public. This is possible at three levels: document, sentence, and feature or aspect [2]. At the document level, the entire document is categorised as neutral, negative, or positive. Sentence level sentiment analysis first divides each sentence into two categories: subjective or objective, and then as positive, negative, or neutral. A sentence is just a small document so there is not a significant difference between the document and sentence levels. Feature level classification is done with respect to words [2].

Human language is difficult for machines to understand, so Natural Language Processing (NLP) is often employed. It is a type of Artificial Intelligence (AI) that facilitates the analysis, interpretation, and evaluation of human understandable data for computer processing. Text processing approaches such as NLP can be used to determine sentiments. Machine learning (ML) techniques have been shown to outperform traditional NLP approaches such as lexicon

based in terms of accuracy [3][4][5]. These results suggest using ML approaches for sentiment analysis.

1.1 Motivation

Sentiment analysis is becoming more important as the amount of information available on social media platforms grows. The related research can be categorised into three primary areas: business, political, and security [6].

From a business point of view, sentiment analysis can provide both employees and customers with recommendations and online advice. This data can also be used to assist ecommerce platforms in analysing their goods and services by revealing consumer preferences. However, the digital nature of online buying makes it difficult to understand and evaluate whether consumers are interested in the feedback or opinions of others.

Sentiment analysis is a key source of political information. People seeking or expressing opinions online are motivated by a variety of factors other than commercial gain. For example, during the two weeks between May 1 and May 14, 2014, more than 1.2 million tweets were collected in three languages (English, French, and German), in order to analyse the debate on Twitter before the European elections [6].

From a public security perspective, socio political events such as the London riots and Arab spring highlight the significance of sentiment analysis in public security [6]. Social media platforms such as Twitter and Facebook were cited in both incidents as important contributors to the development and spread of the events. Authorities can use sentiment analysis to find sensitive material ahead of time. In this case, actions like shutting down Internet communications can deny terrorism followers access to information.

To summarise, sentiment analysis is important not only for consumers and corporations to acquire opinions about products or services, but also for national security and public opinion purposes.

1.2 Related Work

Current tools available for sentiment analysis can deal with a very large amount of customer feedback [12]. In the literature, Twitter sentiment analysis has been used to assist in the discovery of user behaviour. In [8], the Twitter API was used to collect Covid-19 related tweets and determine sentiments about Covid-19. These tweets were then evaluated using ML techniques to determine positive, negative, and neutral sentiments. Latent Dirichlet Allocation (LDA) is an unsupervised learning classifier which finds semantic relationships in groups of words. It has been used to identify the sentiments of tweets. In [9], LDA was used on data extracted from tweets regarding the spread of Covid-19. It was discovered that the majority of tweets on Covid-19 had panic and fear sentiments and those tweets were considered as negative whereas tweets that had trust and comfort sentiments were considered as positive.

Hashtags are single spaced words that begin with the hash symbol (#). In [5], Hashtags related to Covid-19 tweets in the two weeks period from January 14 to January 28, 2020, were extracted and saved as plain text using the Twitter API. Keywords such as infection prevention techniques, vaccination, and racial prejudice were extracted and examined in [7]. Sentiment analysis was then used to determine the positive, negative or neutral sentiments of each tweet using features such as fear, anger, joy, sadness, disgust, and surprise. Finally, unsupervised learning was used to detect features for sentiment analysis.

In [10], the Natural Language Tool Kit (NLTK) was used for sentiment analysis of 53,127 Arabic tweets using the Naive Bayes (NB) classifier. These tweets contained hashtags related to seven government imposed public health initiatives. The results indicated that positive tweets outnumbered negative tweets. In [11], Covid-19 tweets between March 11 and March 31, 2020, were examined. The goal was to determine how people reacted to the disease. The extracted data was noisy and had duplications so preprocessing was employed before classification.

1.3 Report Organization

Chapter 1 presented the problem details and project overview. The related work, motivation, and report structure were discussed.

Chapter 2 gives details on sentiment analysis techniques, NLP, data importing, preprocessing, and the proposed methodology.

Chapter 3 provides an overview of ML, Jupyter notebook, the Python programming language, and ML classifiers.

Chapter 4 presents the performance evaluation of the NB, SVM, RF and DT classifiers in terms of precision, recall, F-measure, accuracy, and execution time. A discussion of the results is also provided.

Chapter 5 gives the conclusion and suggestions for future work.

Chapter 2

Sentiment Analysis

Data is now the most valuable asset and much information about people, from identification to their way of thinking, is available on social media sites. As a result, organisations perform extensive research on this data in order to better understand people and their needs. User opinions, also called sentiments, can be found on all social media sites. Classifiers such as NB, linear regression, RF, DT, and deep learning algorithms such as Convolution Neural Networks (CNNs), Long Short Term Memory (LSTM), and LDA have been used by organisations and researchers to analyze and study sentiments [12]. Reactions to specific incidents can be classified using the results of this analysis. This allows for a better understanding of sentiments.

Sentiment analysis belongs to the broad area of NLP, content analysis, and computational analysis of sentiments. It is used on online discussions and feedback to identify customer perceptions of products, businesses, and services. The study of sentiments has a wide range of applications in areas such as accounting, law, research, entertainment, education, innovation, governmental affairs, and marketing [14]. Social media platforms have given people a way to open up discussions and share their thoughts and views. Customers can express their thoughts and feelings more on social media than ever before [13].

2.1 Social Media

Online networking is a collection of Internet connected apps that allow for the creation and exchange of user generated content [14]. Online activities include image sharing, blogging, social gaming, video sharing, virtual meetings, online shopping, and reviews. Social media provides the ability to collaborate and share information throughout the world with many people simultaneously [14]. Governments use the Internet to keep in touch with citizens and the world. Businesses use it to recruit and retain consumers, promote products, and provide

benefits or support to consumers [12]. Examples of social media platforms for online networking are Twitter, Facebook, YouTube, and Instagram.

2.2 Twitter and Sentiment Analysis

Twitter is a social media platform founded in 2006 that allows users to communicate via tweets [9]. A tweet is up to 280 characters of text. With more than 500 million users and millions of tweets sent every day, Twitter has quickly become a significant resource for businesses to monitor their reputation and brands by analysing public perception regarding items, services, and even competitors [9]. Many organisations and customers use Twitter to share links to interesting articles or data. While identifying specific reasons for Twitter success is difficult, it has firmly established itself as a growing platform for the dissemination of information [15].

2.3 Sentiment Analysis Techniques

Sentiment analysis is an NLP technique which is used to extract polarity and subjectivity from words and phrases [16][17]. Sentiment analysis, machine translation, text classification, and speech recognition are examples of NLP. In this report, an ML based approach to sentiment analysis is used as depicted in Figure 2.1. The Twitter dataset is first extracted using a Python script with a Covid-19 keyword search. Preprocessing is conducted, features are extracted, and the subjectivity and polarity are determined. Finally, classification is performed to obtain the sentiments.

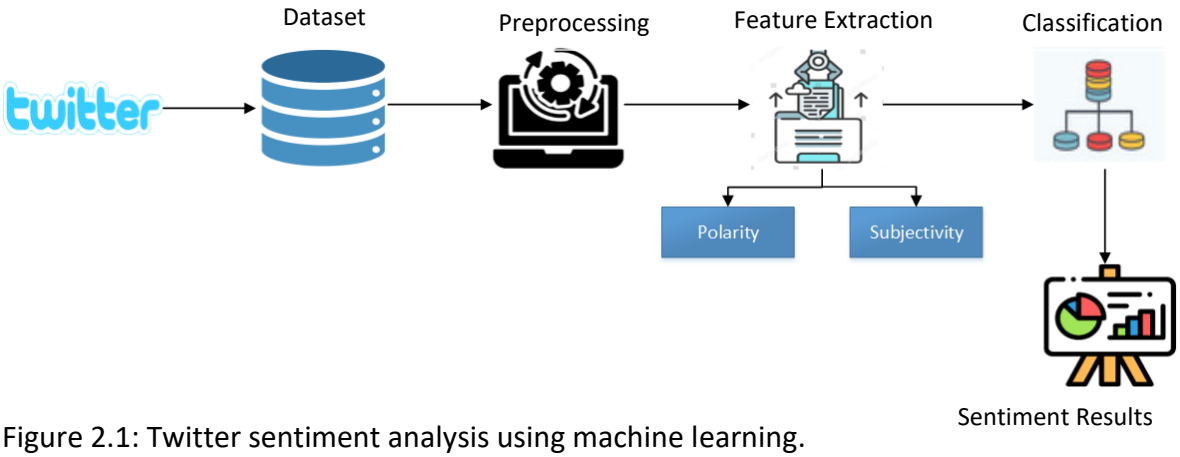


Figure 2.1: Twitter sentiment analysis using machine learning.

2.4 Proposed Framework

The proposed framework is shown in Figure 2.2 and the description of each part is discussed in subsequent sections of this chapter.

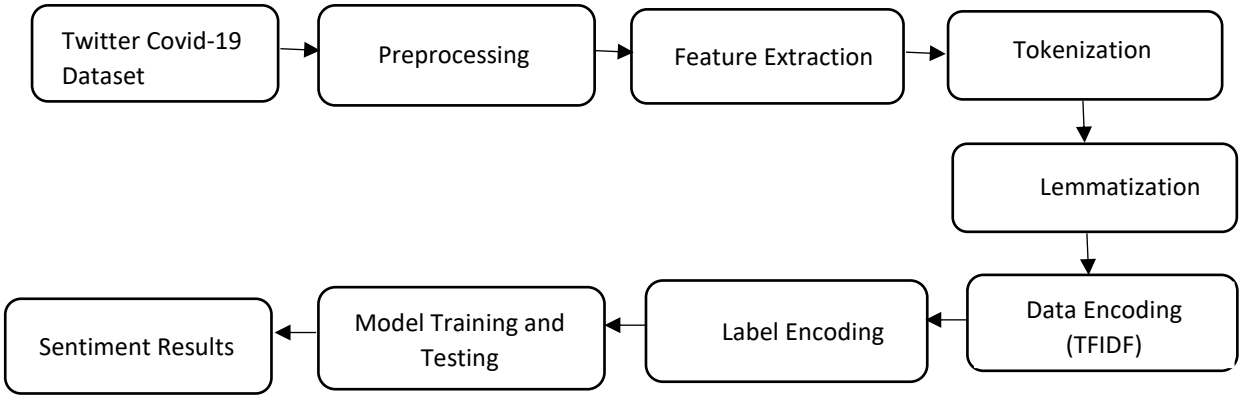


Figure 2.2: The proposed framework.

2.4.1 Twitter Covid-19 Dataset Extraction

Tweets between September and October 2021 were extracted using the Twitter developer account and the Tweepy Python API library was used to select tweets using a keyword search on #Covid-19. 178240 tweets about Covid-19 were obtained and saved in csv format. A word cloud of these tweets is given in Figure 2.3 and contain information such as user_name, user_location, user_description, user_created, user_follower, user_friends, user_favouries, user_verfied, date, text, hashtags, source, and is_retweet, but only a subset of these are useful for sentiment analysis. The Python Pandas library was used to import the csv data into the Jupyter Notebook data frame for processing. Figure 2.4 gives some examples of tweets extracted from Twitter.

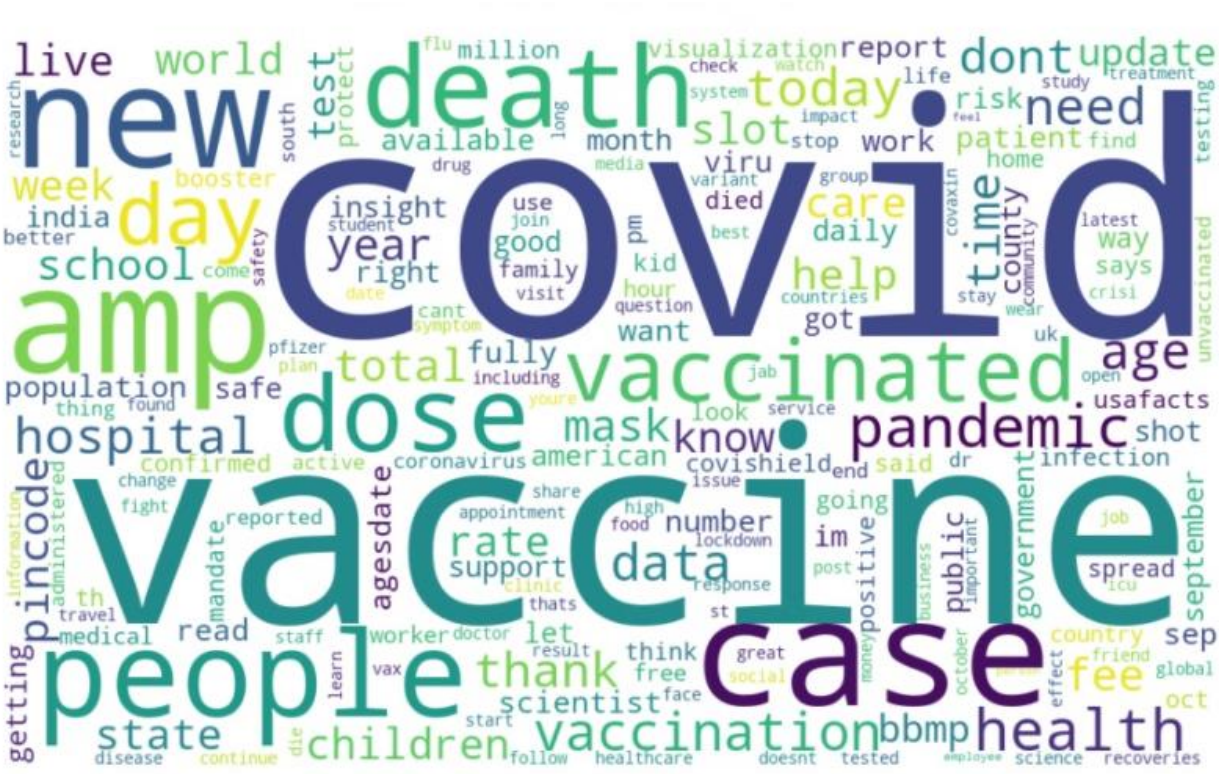


Figure 2.3: Word cloud of the extracted Covid-19 dataset.

Tweet: Has anyone calculated the net decrease in annual payouts by SSA based on daily Covid19 death distribution? #covid19 #data.

Tweet: Daily US Confirmed Covid Cases by County For IA 2021-09-16: Covid visualizations give insight into covid-19 thanks to our data scientists and USAFacts #covidvislualizations #coviddata #smcanalytics #covid19 #coronavirus #covid_19 https://t.co/MVvckMDIhQ.

Tweet: Just because you're stuck inside, that doesn't mean you shouldn't exercise! Try some simplestretches.
#COVID19 #COVID.

Tweet: Given the generally lower hospitalization and death rates from #COVID19 in vaccinated people, it may be more difficult to demonstrate a clinically meaningful benefit to booster shots. Especially if they are given BEFORE immunity wanes significantly.

Figure 2.4: Examples of Twitter Covid-19 tweets.

2.4.2 Preprocessing

Hashtag symbols (#), username (@), URL, RT (symbol for when a tweet is retweeted), emoticons (,), email addresses, quotation marks, and other information are all present in the extracted Twitter data, and are not useful for sentiment analysis. As a result, this data must

be processed in order to create a dataset that classification methods can easily use [18]. Regular expression allows a search for specific patterns in a string of the text. Python has a module called Neattext. It offers several methods for cleaning data including removing hashtags, URLs, retweets, hyperlinks, userhandles, and email addresses. This study used regular expression to identify URLs, such as ((www.[S]+)|(https?://[S]+)) [19], punctuation marks, numbers, and @handle. @handle is a Twitter feature that allows users to reference other users in their tweets. Hashtags are replaced with words without the hashtag symbol, for example, #Covid is replaced with Covid-19.

Stop words are words such as I, we, is, am, are, that, this, a, and an. These words are removed because they carry no useful information. A library in the NLTK module contains a list of stopwords. These stopwords are removed by comparing each word in the dataset to the words in this library.

2.4.3 Feature Extraction

Textblob is a Python library for processing text data and performing NLP tasks such as speech tagging, word extraction, sentiment analysis, classification, and translation. The goal of sentiment analysis is to learn how people are feeling, what they are thinking about situations, and how they have expressed their emotions on social media. The sentiment function of TextBlob returns polarity and subjectivity. Polarity is a value in the range [-1, 1], where > 0 is positive sentiment, 0 is neutral, and negative sentiment is < 0. It is also important to know whether information is a known fact or simply opinion. This is referred to as data subjectivity, and allows researchers to determine what kinds of thoughts influence people. Subjectivity is a value in the range [0, 1] where 0 is a fact and 1 is an opinion [20]. Examples of subjectivity and polarity for tweets in the dataset with neutral, positive, and negative sentiments are given below.

Tweet: Has anyone calculated the net decrease in annual payouts by SSA based on daily Covid19 death distribution? #covid19 #data.
polarity = 0.0, subjectivity = 0.0, sentiment = Neutral
Tweet: Best way protect vaccinated book appointment attend dropin centre visit.
polarity = 1.0, subjectivity = 0.3, sentiment = Positive

Tweet: Making mark patient care pandemic hard times takes patience commitment comes failures test let push inaction learn persevere.
polarity = -0.29, subjectivity = 0.54, sentiment = Negative

Examples of fact and opinion text are given below.

Fact: wearing mask and getting vaccine shots prevent from Covid-19 spread
Subjectivity = 0.0
Opinion: "we're prepared, and we're doing a great job with it. And it will go away. Just stay calm. It will go away." — Trump after meeting with Republican senators [20].
Subjectivity = 0.75

2.4.4 Tokenization

Tokenization is the process of dividing text into tokens (single words) [21]. This is useful because text data contains a wide range of words that make it difficult to analyse. As a result, the data is divided into tokens to make it easier to understand the meaning for sentiment analysis. The following tweet from the dataset is an example of tokenization.
Tweet: daily confirmed covid cases county is Italy
Tokenized: daily, confirmed, covid, cases, county, is, Italy

2.4.5 Lemmatization

Words are commonly used in various forms and tenses all of which have the same meaning but vary in spelling. This makes analysing and processing words difficult so lemmatization is used in data preprocessing [21]. An example of lemmatization of a tweet from the dataset is given below.
Tweet: Isolated remote communities with low vaccinated
Lemmatized: Isolating remote communities with low vaccination

2.4.5 Data Encoding

Extracting the meaning of words from a tweet is a difficult task. Thus, tweets must be encoded in a way that machines can understand and extract meaning from. In this project, Term

Frequency Inverse Document Frequency (TFIDF) is used to translate words into a sequence of numbers. Term Frequency (TF) is the frequency of a word appearing in a tweet divided by the total number of words in the tweet, so each tweet has its own term frequency

$$\text{TF} = \frac{\text{Frequency of a word in a tweet}}{\text{Total number of words in the tweet}}$$

Inverse Document Frequency (IDF) is the log of the number of tweets divided by the number of tweets containing a word

$$\text{IDF} = \log \frac{\text{Number of tweets}}{\text{Number of tweets containing a word}}$$

so TFIDF = TF x IDF

2.4.6 Label Encoding

After preprocessing, the dataset is labelled using the class labels positive, negative, and neutral. If polarity > 0, the label is positive, if polarity < 0 the label is negative, and if polarity = 0, the label is neutral.

2.4.7 Model Training and Testing

In this step, the classifiers are trained and tested using the dataset. The train test split method is used with 70% train and 30% test, 50% train and 50% test, and 70% test and 30% test. To avoid overfitting, the stratified parameter in the train_test Python split method is used. This method uses stratified folds to construct classes with similar distributions.

2.4.8 Sentiment Results

Sentiment results are obtained for the NB, SVM, DT, and RF classifiers. These results are evaluated using accuracy, F-measure, recall, precision, and execution time as metrics.

Chapter 3

Machine Learning

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that allows computers to learn without being programmed explicitly. Classification, clustering, and regression analysis can be done using ML [22]. In most cases, the classifier is trained using a training dataset and then classification or prediction is done using a test dataset. Supervised learning, unsupervised learning, and semi supervised learning are three types of machine learning [22].

Supervised learning is the most extensively used ML technique [22]. In this case, the classifier is trained using a labelled dataset and then it is used for prediction, classification or regression [22]. Supervised learning is employed in a wide variety of NLP applications such as sentiment analysis. Classification can be binary or multi-class. For example, predicting discrete values such as positive, negative, and neutral is a multi-class problem. Regression is used to predict continuous values such as age, salary, and price. NB, SVM, DT, and RF are the supervised machine learning classifiers considered in this report.

With unsupervised learning, a labelled dataset is not used for training. Thus, the classifier learns without any prior knowledge. It identifies important properties or patterns of the dataset and the acquired knowledge is used in prediction and classification. Unsupervised learning is typically used for clustering and dimensionality reduction tasks. Semi supervised learning combines the benefits of supervised and unsupervised learning. It employs both labelled and unlabeled data for training [22].

3.1 ML Tools

3.1.1 Python

Python is an English like programming language which has been used to create classifiers. It is preferred by programmers and researchers as it is open source, easy to learn, simple, interpreted, object oriented, reusable, and modular. It has extensive libraries and also supports packages and modules [27].

3.1.2 Jupyter Notebook

Jupyter Notebook is an open source project that supports interactive data research and scientific computing using a variety of programming languages including Python and MATLAB. [25]. Notebooks are browser based applications used for scientific computing from initial steps such as data importing and preprocessing to visualizing detailed results [26]. The Jupyter Notebook IDE is shown in Figure 3.4. It is divided into cells that can include text, video, images, code, and arithmetic operations. These are combined to create an interactive document. Jupyter Notebook contains a menu bar and a tool bar. These bars can be used to add and delete comments and cells and a notebook can be saved.

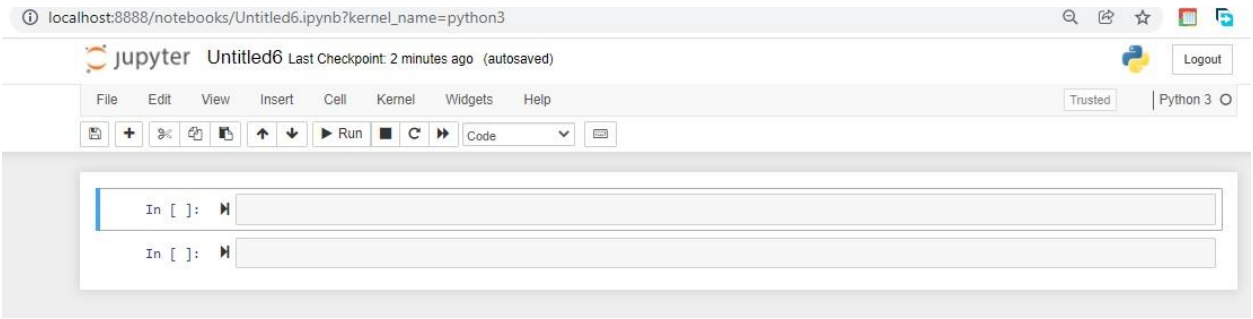


Figure 3.1: The Jupyter Notebook IDE.

3.2 Machine Learning Classifiers

3.2.1 Naive Bayes (NB)

NB is a simple classifier used for NLP problems. It is a supervised ML techniques which is particularly useful for text classification. NB calculates the posterior probability of a class

based on the distribution of the words (features) in a dataset. It uses Bayes' Theorem to estimate the likelihood of a given feature [23]. Bernoulli NB is used for multiple features with binary values and multinomial NB is used for text classification.

3.2.2 Support Vector Machine (SVM)

SVM is a powerful supervised ML classifier that can be used for both classification and regression. However, it is primarily used in classification problems [16]. The idea behind SVM is to discover linear separators or hyperplanes in the search space that can best separate the classes. The hyperplane that provides the widest separation with the largest normalized distance between data points is used [23]. SVM is ideal for text classification as the sparse nature of text means few features are unimportant and they tend to be associated with one another [24].

3.2.3 Decision Tree (DT)

DT has a hierarchical structure. Like an inverted tree, it starts from a root node and moves downwards to branches and then leaves. It can be used for classification and regression problems. Each node is labelled with a condition and features are split based on the conditions at the nodes. This process continues until a leaf node is reached, at which point the result is predicted. DT operates on the basis of independent variables. The Gini index is used to classify the data, and the highest value attributes are selected for the next iteration [24].

3.2.4 Random Forest (RF)

RF is an ML classifier that aggregates the outputs of several DTs applied to distinct subsets of a dataset to improve prediction accuracy. A condition is applied to one or more features of the data at each tree node. RF results are obtained by combining predictions from several trees [24]. Each tree votes for a specific class and the predicted class is the one with the most votes. RF has been shown to have good accuracy with unbalanced dataset [24].

Chapter 4

Results and Discussion

The sentiment analysis of 178241 tweets from the Covid-19 dataset is presented in this chapter using the supervised ML classifiers NB, SVM, DT, and RF. The performance metrics used are accuracy, precision, recall, F-measure, and execution time. Undersampling and oversampling are used to improve the performance. The number of neutral, positive, and negative tweets in the Covid-19 dataset is given in Figure 4.1. This shows there are 75630 neutral, 70947 positive, and 31664 negative tweets.

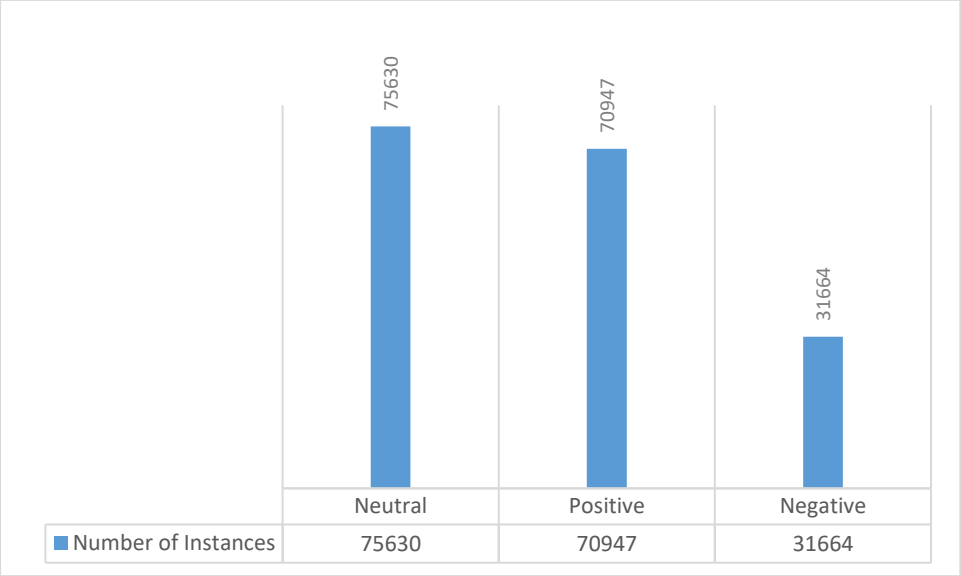


Figure 4.1: The number of tweets in the Covid-19 dataset.

Undersampling and oversampling of the dataset is used to avoid overfitting. Undersampling is a simple approach which randomly undersamples classes with more instances. Synthetic Minority Oversampling Technique (SMOTE) is a statistical method which is used to increase the number of instances in the minority (smallest) classes. It creates new instances based on existing instances [28]. To balance the number of tweets, undersampling was used on the neutral and positive tweets and oversampling (SMOTE) on the negative and positive tweets.

The number of tweets in the balanced dataset after undersampling is shown in Figure 4.2, and after oversampling in Figure 4.3. Figure 4.4 shows that a subset was created by randomly selecting 10000 tweets from each of the neutral, positive, and negative classes.

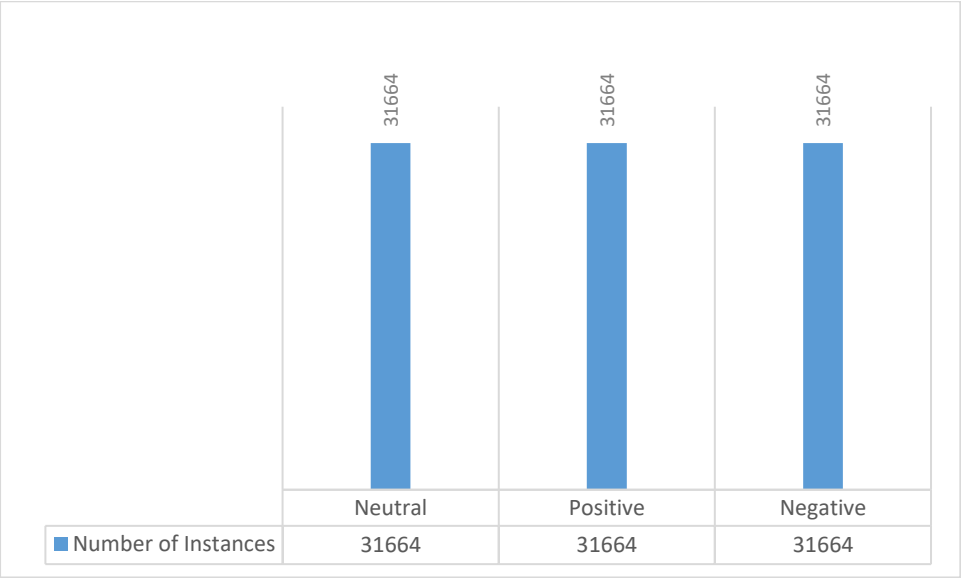


Figure 4.2: The number of tweets in the Covid-19 dataset after undersampling.

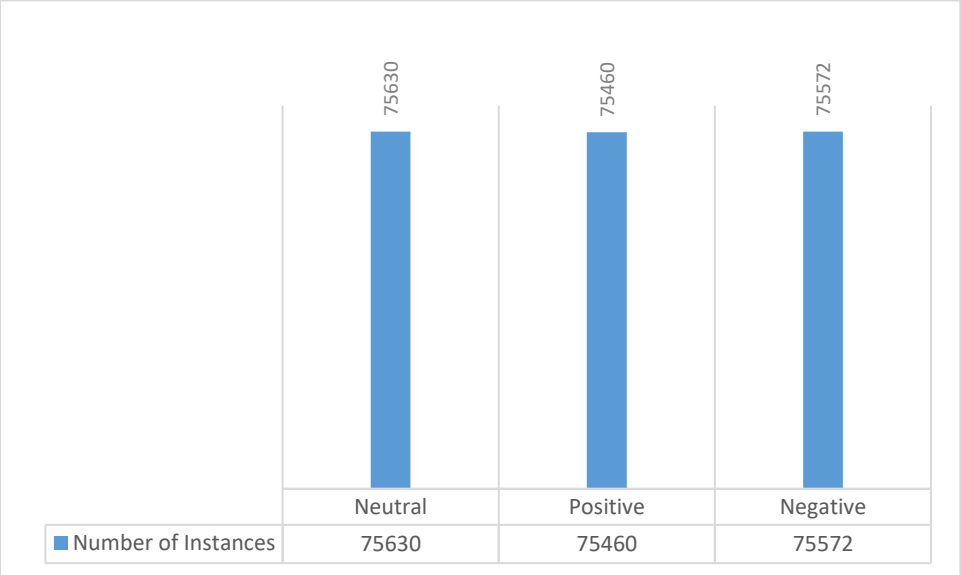


Figure 4.3: The number of tweets in the Covid-19 dataset after oversampling.

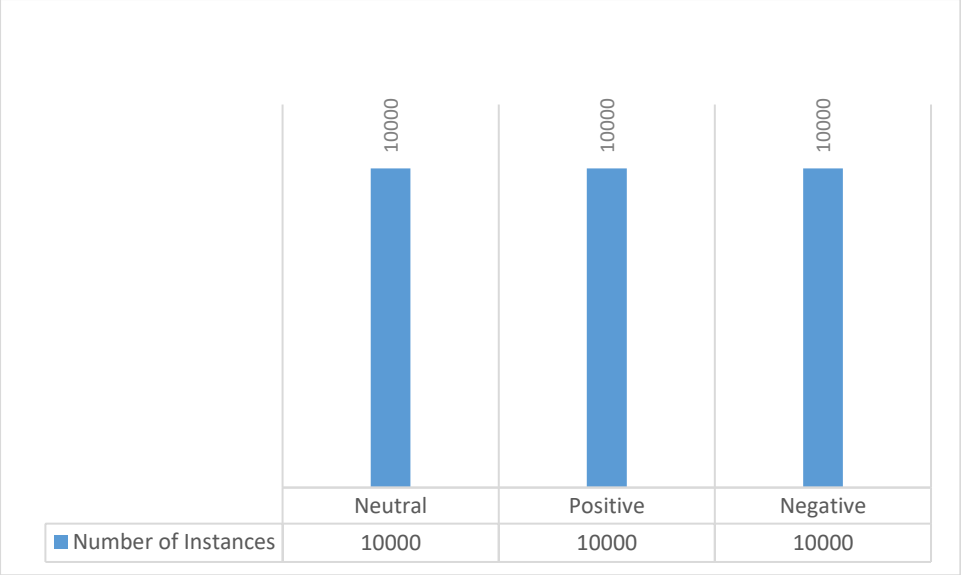


Figure 4.4: The 10000 tweets in each class Covid-19 dataset.

4.1 Evaluation Metrics

Accuracy is commonly used to assess classifier performance. It is the ratio of the number of properly classified samples to the total number of samples. Automatic sentiment analysis should have high accuracy in predicting whether sentiments are positive, negative, or neutral. For imbalanced datasets, it is beneficial to evaluate the model performance using other metrics [30]. For binary classification, recall is defined as the ratio of correctly predicted positive samples to the total number of positive samples and F-measure is the weighted average of precision and recall [30]. However, for multiclass classification, F-measure should consider the recall and precision for each class. To do this, the positive, neutral, and negative class parameters in the confusion matrix in Table 4.1 can be employed. In this table, R_1C_1 denotes predicted neutral and actually neutral.

	Neutral	Positive	Negative
Neutral	R_1C_1	R_1C_2	R_1C_3
Positive	R_2C_1	R_2C_2	R_2C_3
Negative	R_3C_1	R_3C_2	R_3C_3

Table 4.1: Confusion matrix for neutral, positive and negative classes.

Accuracy is the ratio of the sum of correctly predicted samples to the sum of all samples

$$\textbf{Accuracy} = \frac{R_1C_1 + R_2C_2 + R_3C_3}{R_1C_1 + R_1C_2 + R_1C_3 + R_2C_1 + R_2C_2 + R_2C_3 + R_3C_1 + R_3C_2 + R_3C_3}$$

where R_1C_1 is the number of correctly classified neutral sentiments as neutral, R_2C_2 is the number of correctly classified positive sentiments as positive, R_3C_3 is the number of correctly classified negative sentiments as negative, and R_1C_2, R_1C_3, R_2C_1, R_2C_3, R_3C_1, R_3C_2 are the numbers of misclassified sentiments.

Precision is the number of tweets correctly predicted for each class divided by the number of predicted samples for the class

$$\textbf{Precision}_{\text{neutral}} = \frac{R_1C_1}{R_1C_1 + R_1C_2 + R_1C_3}$$

$$\textbf{Precision}_{\text{positive}} = \frac{R_2C_2}{R_2C_1 + R_2C_2 + R_2C_3}$$

$$\textbf{Precision}_{\text{negative}} = \frac{R_3C_3}{R_3C_1 + R_3C_2 + R_3C_3}$$

Recall is the number of tweets correctly predicted for a class divided by the number of actual samples for the class

$$\textbf{Recall}_{\text{neutral}} = \frac{R_1C_1}{R_1C_1 + R_2C_1 + R_3C_1}$$

$$\textbf{Recall}_{\text{positive}} = \frac{R_2C_2}{R_1C_2 + R_2C_2 + R_3C_2}$$

$$\textbf{Recall}_{\text{neutral}} = \frac{R_3C_3}{R_1C_3 + R_3C_2 + R_3C_3}$$

F-measure is the harmonic mean of precision and recall

$$\textbf{F-measure}_{\text{neutral}} = \frac{2R_1C_1}{2\,R_1C_1 + R_1C_2 + R_1C_3 + R_2C_1 + R_3C_1}$$

$$\textbf{F-measure}_{\text{positive}} = \frac{2R_2C_2}{2\,R_2C_2 + R_2C_1 + R_2C_3 + R_1C_2 + R_3C_2}$$

$$\textbf{F-measure}_{\text{negative}} = \frac{2R_3C_3}{2\,R_3C_3 + R_3C_1 + R_3C_2 + R_1C_3 + R_2C_3}$$

The overall precision and recall are the corresponding arithmetic means

$$\textbf{Precision}_{\text{Average}} = \frac{\text{Precision}_{\text{neutral}} + \text{Precision}_{\text{positive}} + \text{Precision}_{\text{negative}}}{3}$$

$$\textbf{Recall}_{\textbf{Average}} = \frac{\text{Recall}_{\text{neutral}} + \text{Recall}_{\text{positive}} + \text{Recall}_{\text{negative}}}{3}$$

$$\textbf{F-measure}_{\textbf{Average}} = \frac{\text{F}-\text{measure}_{\text{neutral}} + \text{F}-\text{measure}_{\text{positive}} + \text{F}-\text{measure}_{\text{negative}}}{3}$$

Training time is the time taken to train the classifier using the Covid-19 dataset.

Testing time is the time taken to test the classifier using the Covid-19 dataset.

In this report, accuracy, F-measure, recall, and precision results are given as percentages, and time in seconds (s).

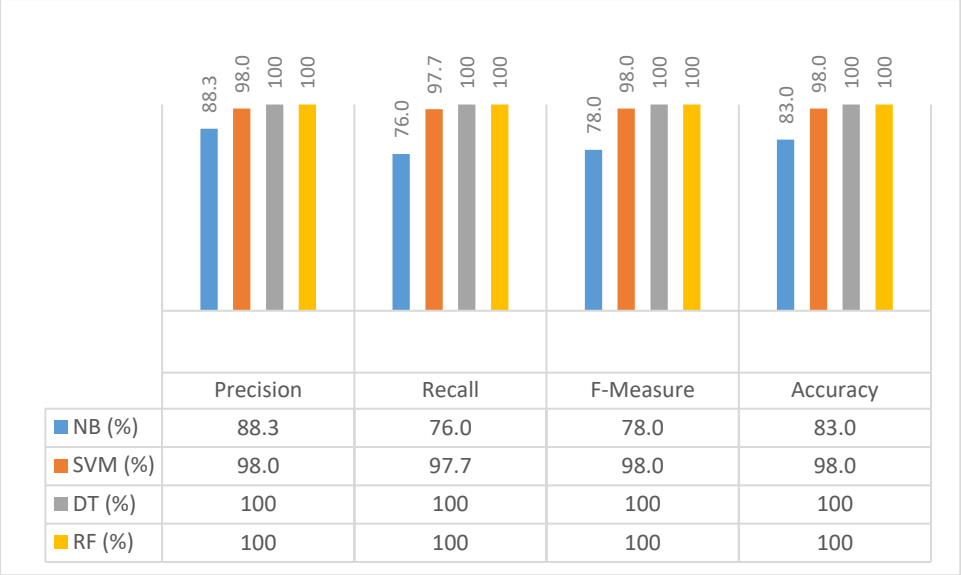
4.2 Classifiers Results

The training and testing results with the NB, SVM, DT, and RF classifiers are given in this section. They are the average of three trials.

4.2.1 Performance without Undersampling and Oversampling with 70-30 Train/Test Split

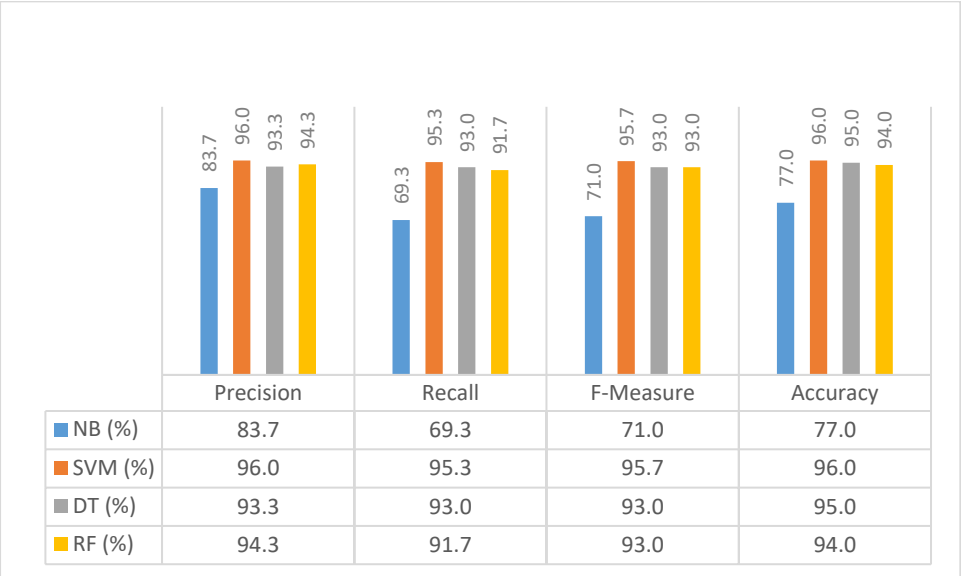
The dataset without undersampling and oversampling is considered in this section. The training results for the ML classifiers are given in Figure 4.5. This shows the accuracy, F-measure, recall and precision with DT and RF are the highest, followed by SVM and then NB. NB has the shortest training time at 0.04 s, followed by DT at 69.1 s, RF at 1178 s, and SVM at 3587 s. There was little variation in time between DT and RF as observed in Figure 4.7, so the former offers the best tradeoff in terms of performance.

Figure 4.6 shows the testing performance of the ML classifiers. The accuracy, F-measure, recall, and precision are the highest with SVM, but it has the highest testing time. The performance of DT and RF is similar, but the former has a lower testing time as shown in Figure 4.7. In terms of testing time NB is the best. The overall testing performance of DT is the best.



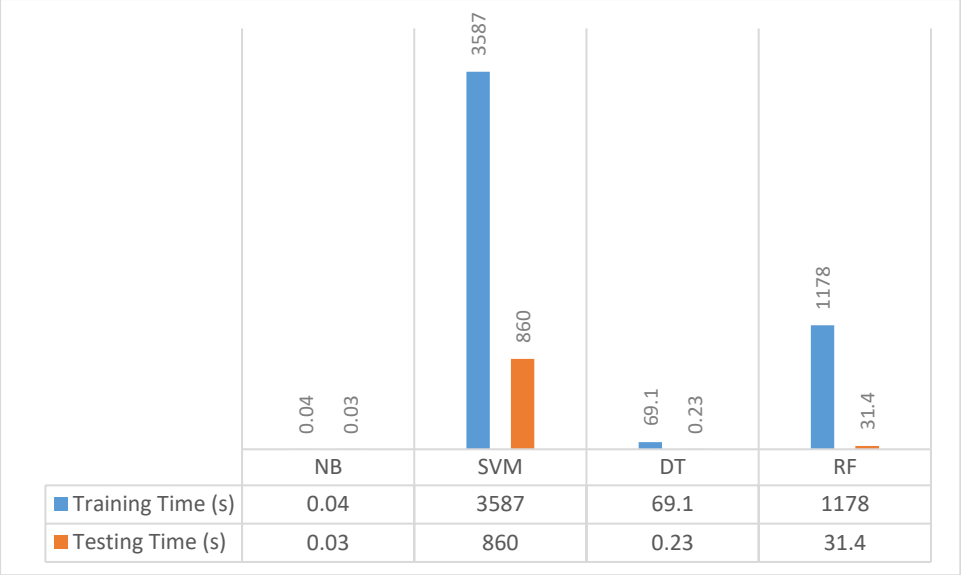
	Precision	Recall	F-Measure	Accuracy
NB (%)	88.3	76.0	78.0	83.0
SVM (%)	98.0	97.7	98.0	98.0
DT (%)	100	100	100	100
RF (%)	100	100	100	100

Figure 4.5: Training results without undersampling or oversampling.



	Precision	Recall	F-Measure	Accuracy
NB (%)	83.7	69.3	71.0	77.0
SVM (%)	96.0	95.3	95.7	96.0
DT (%)	93.3	93.0	93.0	95.0
RF (%)	94.3	91.7	93.0	94.0

Figure 4.6: Testing results without undersampling or oversampling.

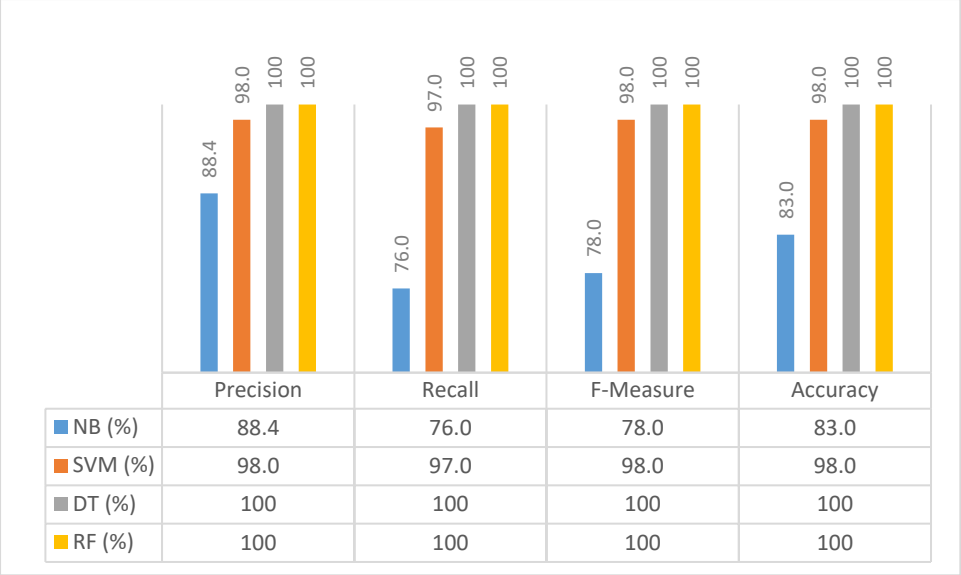


	NB	SVM	DT	RF
Training Time (s)	0.04	3587	69.1	1178
Testing Time (s)	0.03	860	0.23	31.4

Figure 4.7: Training and testing times without undersampling or oversampling.

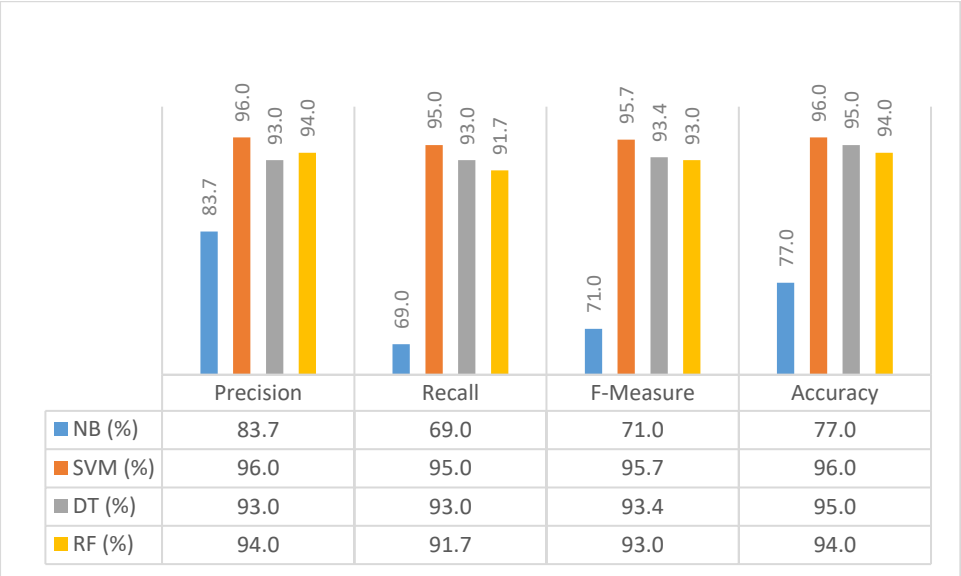
4.2.2 Performance with Undersampling and 70-30 Train/Test Split

Undersampling was used to balance the data by randomly selecting tweets from the majority classes neutral and positive. Resample, a Python sampling method, was employed to randomly select tweets. Undersampling has the drawback of potentially removing important data. The training and testing results using undersampling for the ML classifiers are shown in Figures 4.8, 4.9 and 4.10. The accuracy, F-measure, recall, and precision for NB training are the lowest among all classifiers, but it has the lowest training time. SVM has the highest training time. The RF and DT training results are very good in terms of accuracy, F-measure, recall and precision, but the training time for DT is the best. Figures 4.9 and 4.10 show that SVM has the highest testing accuracy, F-measure, recall, and precision but the highest testing time. NB has the lowest time, but in terms of overall performance DT is the best.



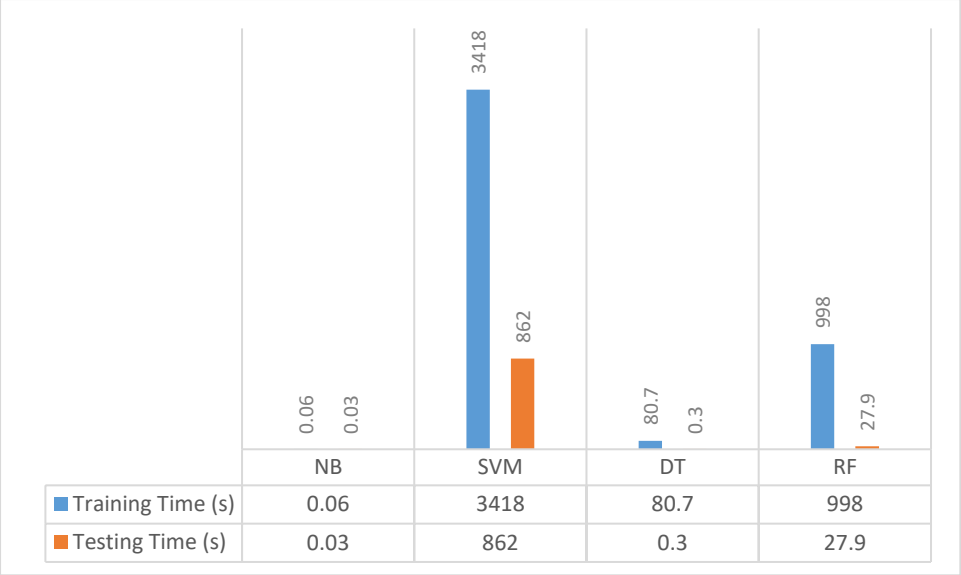
	Precision	Recall	F-Measure	Accuracy
NB (%)	88.4	76.0	78.0	83.0
SVM (%)	98.0	97.0	98.0	98.0
DT (%)	100	100	100	100
RF (%)	100	100	100	100

Figure 4.8: Training results with undersampling.



	Precision	Recall	F-Measure	Accuracy
NB (%)	83.7	69.0	71.0	77.0
SVM (%)	96.0	95.0	95.7	96.0
DT (%)	93.0	93.0	93.4	95.0
RF (%)	94.0	91.7	93.0	94.0

Figure 4.9: Testing results with undersampling.



	NB	SVM	DT	RF
Training Time (s)	0.06	3418	80.7	998
Testing Time (s)	0.03	862	0.3	27.9

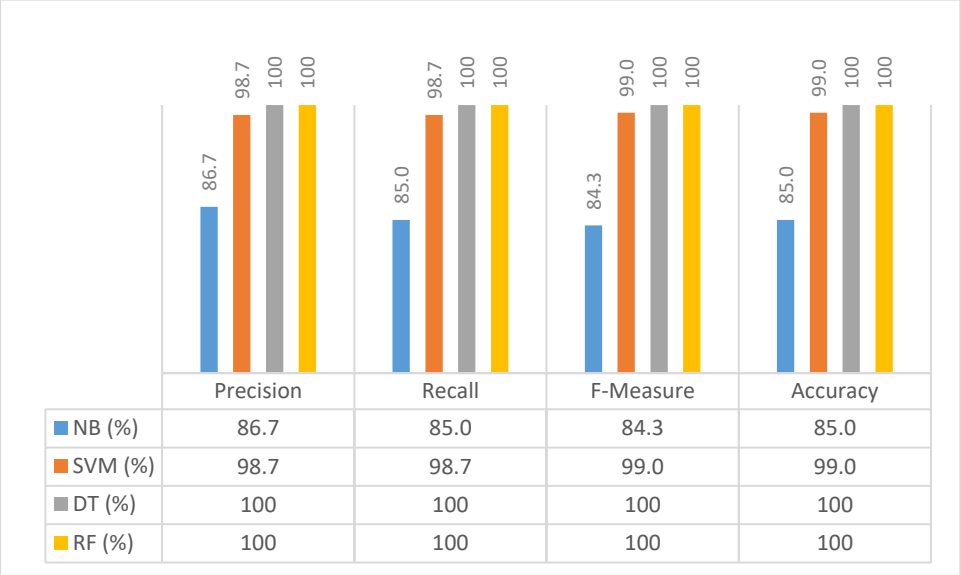
Figure 4.10: Training and testing times with undersampling.

4.2.3 Performance with Oversampling Using SMOTE and 70-30 Train/Test Split

Oversampling is now used to balance the dataset. This can be achieved by duplicating the number of tweets in the minority classes, but this can lead to overfitting. To address this problem, the Synthetic Minority Oversampling Technique (SMOTE) introduced in [28] is used here. It has been shown to be effective in a wide range of situations. It uses feature similarity between instances to produce new synthetic instances [29]. To build a synthetic instance, the k-nearest neighbours from minority instances are selected and linear interpolation is employed to create a new instance between them. SMOTE has two parameters, random_state and k_neighbours. In this report, random_state is equal to 100 and k_neighbours is equal to 1. Figure 4.3 shows the number of dataset tweets after oversampling, and the classifier training and testing performance is shown in Figures 4.11, 4.12, and 4.13. The training accuracy, F-measure, recall, and precision for DT are the best followed by RF, while NB has the worst performance. In addition, NB has the smallest training time of 0.1 s followed by DT at 84.5 s, while SVM has the highest training time of 4699 s followed by RF at 1067 s.

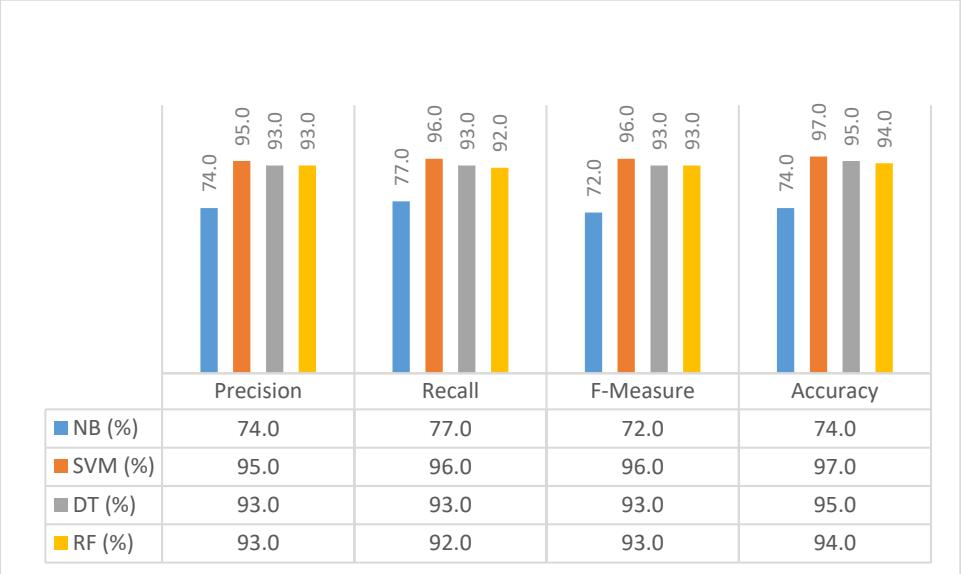
Figure 4.12 shows that SVM accuracy, F-measure, recall, and precision are better than NB, DT, and RF. The training and testing times for SVM are the highest, whereas NB has the smallest

training and testing times. The RF and DT performance is better, but the latter has the best overall performance because DT has lower execution time and better accuracy and recall.



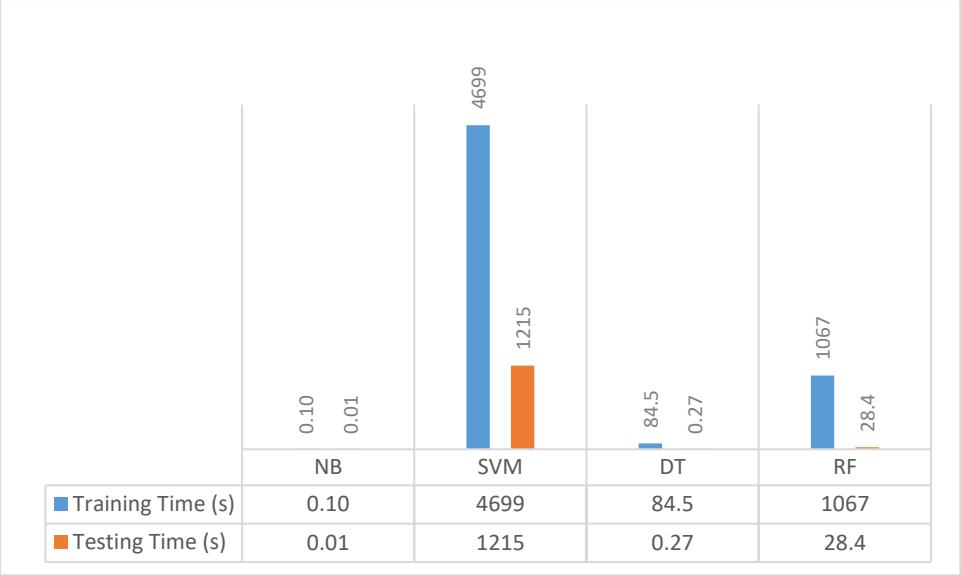
	Precision	Recall	F-Measure	Accuracy
NB (%)	86.7	85.0	84.3	85.0
SVM (%)	98.7	98.7	99.0	99.0
DT (%)	100	100	100	100
RF (%)	100	100	100	100

Figure 4.11: Training results with oversampling using SMOTE.



	Precision	Recall	F-Measure	Accuracy
NB (%)	74.0	77.0	72.0	74.0
SVM (%)	95.0	96.0	96.0	97.0
DT (%)	93.0	93.0	93.0	95.0
RF (%)	93.0	92.0	93.0	94.0

Figure 4.12: Testing results with oversampling using SMOTE.

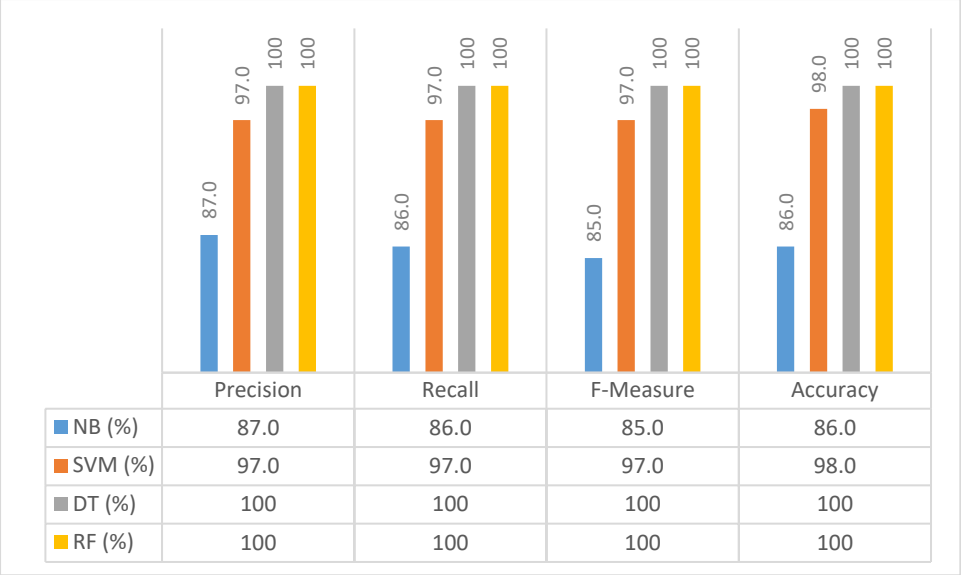


	NB	SVM	DT	RF
Training Time (s)	0.10	4699	84.5	1067
Testing Time (s)	0.01	1215	0.27	28.4

Figure 4.13: Training and testing times with oversampling using SMOTE.

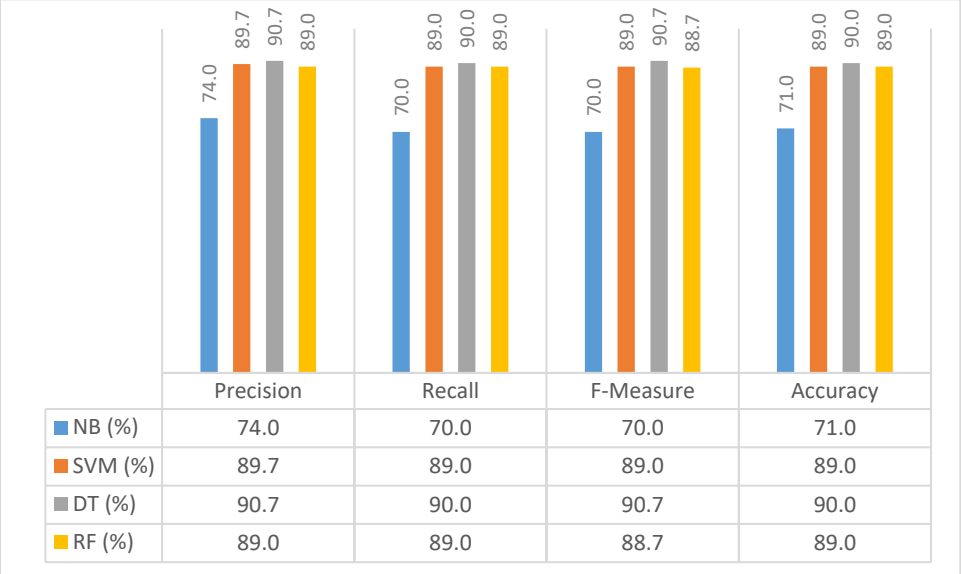
4.2.4 Performance with 10000 Tweets in each Class and 70-30 Train/Test Split

The performance without undersampling and oversampling with 10000 tweets from each class is now given for the split 70% for training and 30% for testing. Figures 4.14, 4.15, and 4.16 give the training and testing results. The training accuracy, F-measure, recall, and precision for RF and DT are the best followed by SVM and then NB. The testing accuracy, F-measure, recall, and precision for DT are the best followed by RF and then SVM. These results show that decreasing the number of tweets in the dataset reduces the training and testing times. NB has the lowest training time of 0.02 s, while SVM has the highest time at 151 s followed by RF at 133 s. Similarly, the testing time for NB is the lowest at 0.01 s and SVM has the highest testing time at 60.7 s. DT is the best in terms of both performance and time.



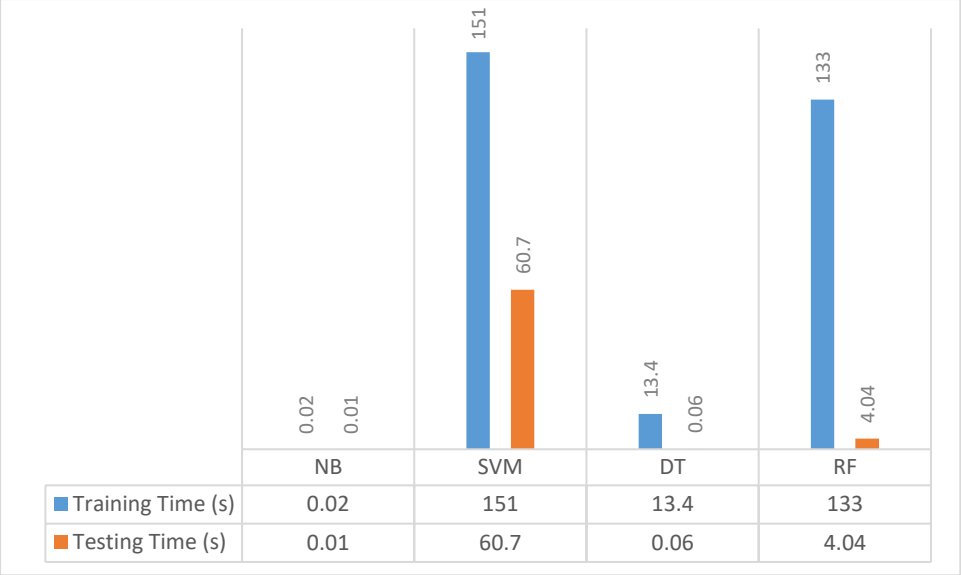
	Precision	Recall	F-Measure	Accuracy
NB (%)	87.0	86.0	85.0	86.0
SVM (%)	97.0	97.0	97.0	98.0
DT (%)	100	100	100	100
RF (%)	100	100	100	100

Figure 4.14: Training results for 10000 tweets in each class.



	Precision	Recall	F-Measure	Accuracy
NB (%)	74.0	70.0	70.0	71.0
SVM (%)	89.7	89.0	89.0	89.0
DT (%)	90.7	90.0	90.7	90.0
RF (%)	89.0	89.0	88.7	89.0

Figure 4.15: Testing results for 10000 tweets in each class.

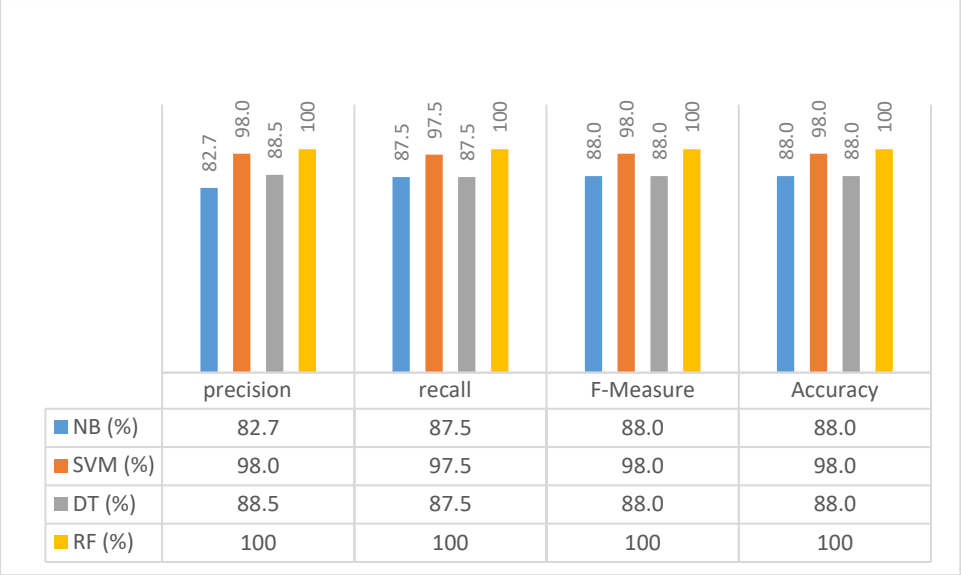


	NB	SVM	DT	RF
Training Time (s)	0.02	151	13.4	133
Testing Time (s)	0.01	60.7	0.06	4.04

Figure 4.16: Training and testing times for 10000 tweets in each class.

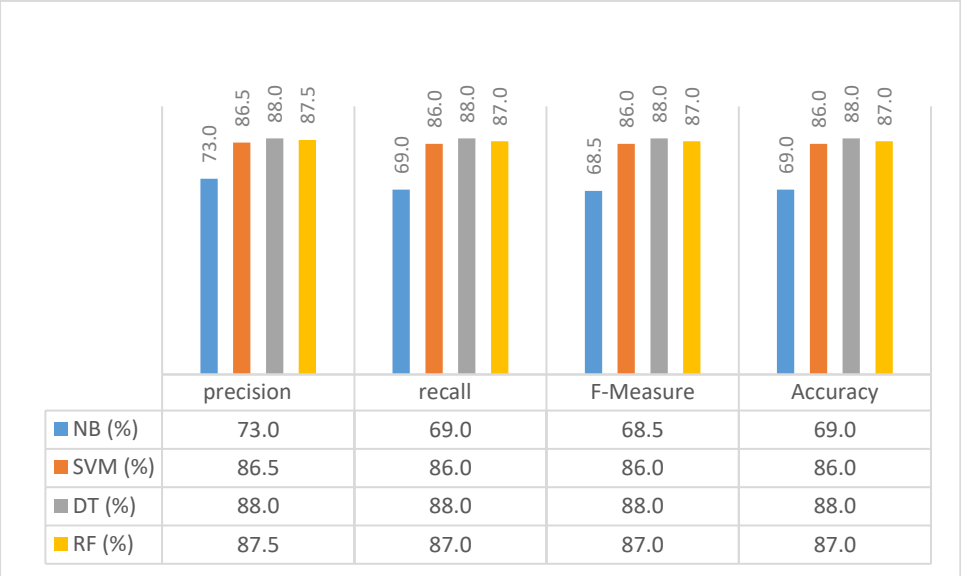
4.2.5 Performance with 10000 Tweets in each Class and 50-50 Train/Test Split

The random selection of 10000 tweets is now considered with a split of 50% for training and 50% for testing. Figure 4.17 gives the training results, Figure 4.18 gives the testing results, and Figure 4.19 gives the training and testing times. These results shows that RF has the best training accuracy, F-measure, recall, and precision, followed by SVM. The training accuracy, F-measure, and recall of DT and NB are similar. For testing, DT is the best in terms of accuracy, F-measure, recall, and precision followed by RF, whereas NB is the worst. The training time of RF is the highest as shown in Figure 4.19 followed by SVM. In terms of training and testing times, NB is the best among the classifiers followed by DT. Overall, DT has the best testing results.



	precision	recall	F-Measure	Accuracy
NB (%)	82.7	87.5	88.0	88.0
SVM (%)	98.0	97.5	98.0	98.0
DT (%)	88.5	87.5	88.0	88.0
RF (%)	100	100	100	100

Figure 4.17: Training results for 10000 tweets in each class with 50-50 train/test split.



	precision	recall	F-Measure	Accuracy
NB (%)	73.0	69.0	68.5	69.0
SVM (%)	86.5	86.0	86.0	86.0
DT (%)	88.0	88.0	88.0	88.0
RF (%)	87.5	87.0	87.0	87.0

Figure 4.18: Testing results for 10000 tweets in each class with 50-50 train/test split.

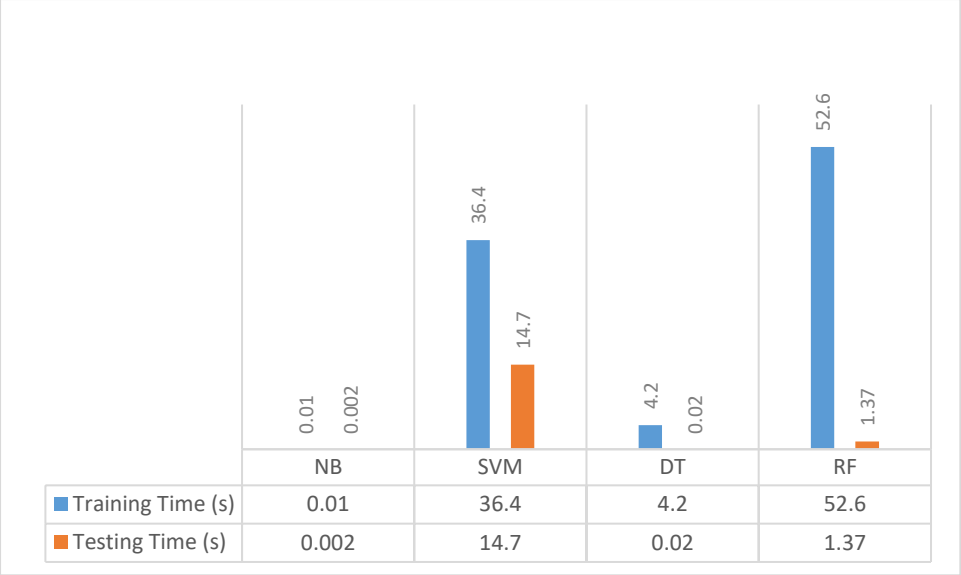
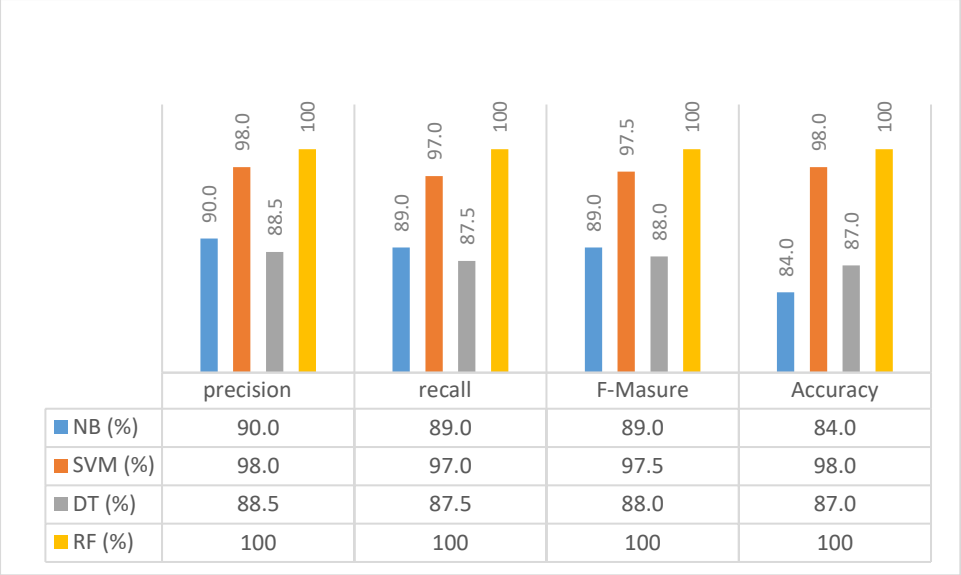


Figure 4.19: Training and testing times for 10000 tweets in each class with 50-50 train/test split.

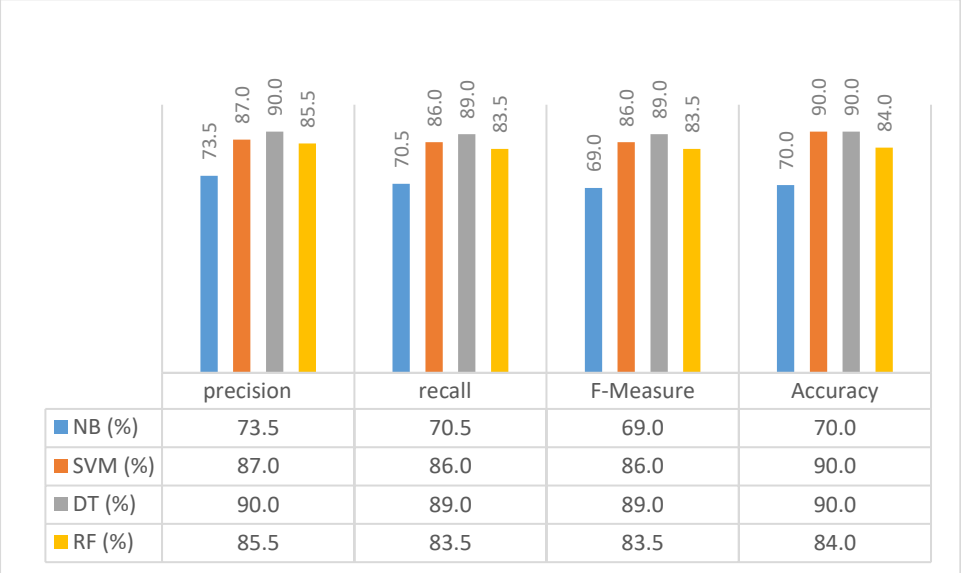
4.2.6 Performance with 10000 Tweets in each Class and 30-70 Train/Test Split

In this section, a 30% training and 70% testing split of the 10000 tweets in each class dataset is considered. Figure 4.20 shows that RF has the best training performance in terms of accuracy, F-measure, recall, and precision followed by SVM. However, SVM has the highest training time followed by RF. The accuracy with DT is better than NB, but NB has a smaller training time. Figure 4.21 gives the testing accuracy, F-measure, recall, and precision and Figure 4.22 gives the training and testing times. These result shows that DT is the best in terms of testing accuracy, F-measure, recall, and precision followed by SVM and RF. NB is the best in terms of training and testing times whereas SVM has the highest training and testing times. Overall, DT has the best testing results in terms of accuracy, F-measure, recall, precision, and execution time.



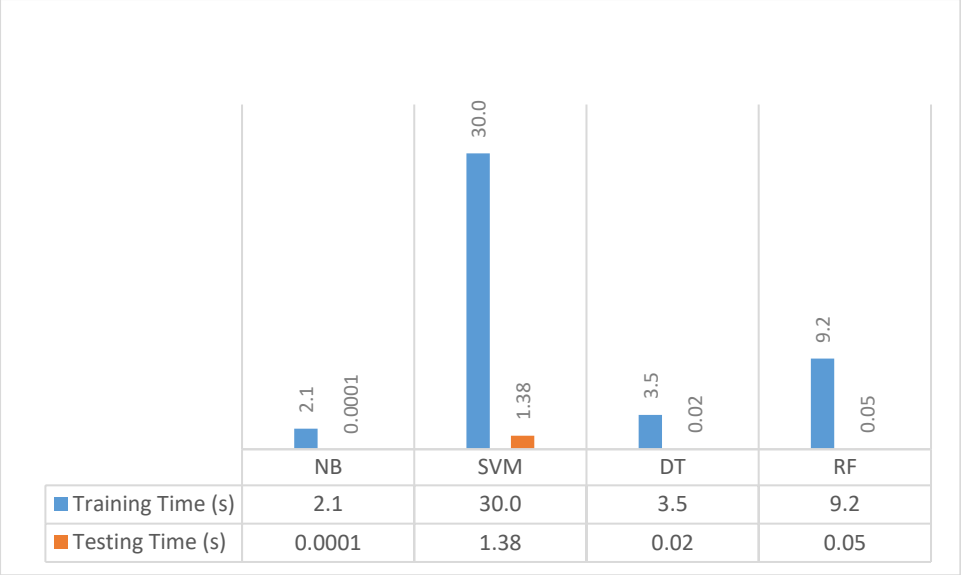
	precision	recall	F-Masure	Accuracy
NB (%)	90.0	89.0	89.0	84.0
SVM (%)	98.0	97.0	97.5	98.0
DT (%)	88.5	87.5	88.0	87.0
RF (%)	100	100	100	100

Figure 4.20: Training results for 10000 tweets in each class with 30-70 train/test split.



	precision	recall	F-Measure	Accuracy
NB (%)	73.5	70.5	69.0	70.0
SVM (%)	87.0	86.0	86.0	90.0
DT (%)	90.0	89.0	89.0	90.0
RF (%)	85.5	83.5	83.5	84.0

Figure 4.21: Testing results for 10000 tweets in each class with 30-70 train/test split.



	NB	SVM	DT	RF
Training Time (s)	2.1	30.0	3.5	9.2
Testing Time (s)	0.0001	1.38	0.02	0.05

Figure 4.22: Training and testing times for 10000 tweets in each class with 30-70 train/test split.

4.3 Discussion

The results of the classifiers without oversampling and oversampling and 70-30 train/test split shows that the accuracy, F-measure, recall, and precision of DT and RF are the highest, followed by SVM and then NB. NB is the fastest in training at 0.04 s whereas SVM is the slowest at 3587 s. DT has the best overall testing results.

With undersampling and 70-30 train/test split, DT and RF have the best training accuracy, F-measure, recall, and precision whereas the time taken to train DT and RF is 80.7 s and 998 s respectively. The testing accuracy, F-measure, recall, and precision for SVM are the highest at 96.0%, 95.0%, 95.7%, and 96.0%, respectively, but the testing time is longest at 3418 s.

With oversampling using SMOTE and 70-30 train/test split, the training accuracy, F-measure, recall, and precision for DT and RF are the best. The DT training time is 84.50 s whereas the RF training time is 1067 s. The testing accuracy, F-measure, recall, and precision of SVM are 95.0%, 96.0%, 96.0% and 97.0%, respectively, which are the best whereas the testing time of SVM and NB are 1215 s and 0.01 s, respectively.

The DT and RF training results for 10000 tweets in each class and 70-30 train/test split are the best in terms of accuracy, F-measure, recall, and precision. The testing accuracy, F-measure, recall, and precision of DT are the highest at 90.0%, 90.7%, 90.0%, and 90.7%, respectively.

The NB training and testing times are the lowest at 0.02 s and 0.01 s, respectively, whereas SVM has the longest training and testing times at 151 s and 60.7 s, respectively. In terms of training and testing accuracy, F-measure, recall, precision, and time, DT is the best.

For 10000 random tweets in each class and 50-50 train/test split, RF training accuracy, F-measure, recall, and precision are the best at 100.0 % followed by SVM accuracy, F-measure, and precision at 98.0%, and recall at 87.50%. Training time for RF is 52.60 s and for SVM is 36.40 s. The testing accuracy, F-measure, recall and precision of DT are 88.0% followed by RF. The DT training time is 4.20 s and testing time is 0.02 s.

For 10000 random tweets in each class and 30-70 train/test split, the training accuracy, F-measure, recall, and precision of RF are the best at 100.0% followed by SVM at 98.0%, 97.5%, 97.0%, and 98.0%. The testing accuracy of DT and SVM is the best at 90.0% but the F-measure, recall, and precision of DT are the best at 89.0%, 89.0% and 90.0%, respectively. The DT testing time is 0.02 s whereas for SVM it is 1.38 s.

Without oversampling and undersampling, with oversampling, with undersampling, and with 10000 tweets and a 70-30 train/test split, the training results of DT and RF in terms of accuracy, F-measure, recall, and precision are similar. DT took 982.5 s less training time with oversampling, 917.3 s less with undersampling, 1109 s less without oversampling and undersampling, and 119.6 s less with 10000 tweets than RF. The training accuracy of SVM with oversampling is 14.0% more than NB, but the SVM training time is 4698.9 s more than NB with oversampling. The testing accuracy of DT with 10000 tweets is 1.0% more than RF and SVM, whereas with oversampling the SVM testing accuracy is 2.0% more than DT and 21.0% more than NB.

The training results with 10000 tweets in each class and a 50-50 train/test split show that the RF accuracy, precision, and recall is 2.0% more than SVM and 12.0% more than DT, whereas with a 30-70 train/test split RF accuracy and precision is 2.0% more than SVM and 13.0% more than DT. RF took 16.2 s more than SVM with a 50-50 train/test split and 48.4 s more than DT, but with a 30-70 train/test split RF took 22.8 s less than SVM and RF took 5.7 s more than DT. It was also observed that the NB accuracy, F-measure, recall, and precision training results with 50-50 and 30-70 train/test splits gave 12.0-19.0% better results than with a 70-30 train/test split. DT testing F-measure and recall with a 30-70 train/test split is 3.0% more than

SVM and RF testing accuracy is 6.0% less than DT. The DT testing accuracy, F-measure, and recall with a 50-50 train/test split is 1.0% more than RF and 2.0% more than SVM. SVM has a testing time 14.68 s more than DT and 13.3 s more than RF.

Overall, the training results of DT and RF are the best in terms of accuracy, F-measure, recall and precision using the train test split method without oversampling and undersampling, with oversampling using SMOTE, with undersampling, and with 10000 tweets in each class. The training and testing times for NB are the lowest with the lowest accuracy and SVM has the highest times with the best test accuracy. In terms of overall results, DT is the best in most of the scenarios considered. It was also observed that a smaller number of tweets reduced the accuracy, F-measure, recall, precision and execution time.

Chapter 5

Conclusion and Future Work

Machine Learning (ML) classifiers were used for sentiment analysis prediction of Twitter tweets related to Covid-19. Three sentiment classes, namely positive, negative, and neutral, were considered and four supervised ML classifiers were used, namely NB, SVM, RF, and DT. Several train/test splits of the dataset were employed. Accuracy, F-measure, recall, precision, and execution time were considered for performance evaluation. The results obtained showed that execution time for SVM is the highest, NB had the fastest training and testing times, but the worst accuracy, F-measure, recall, and precision. Overall, DT provided the best tradeoff between execution time and performance.

5.1 Future Work

In this study, four supervised ML classifiers were used with train/test split of the dataset. Other ML classifiers can be considered in the future with k-fold cross-validation. DL classifiers such as LSTM and LDA could also be used. A comparative analysis using the Waikato Environment for Knowledge Analysis (WEKA) can be conducted as it supports ML classifiers in NLP. WEKA is a Graphical User Interface (GUI) written in the Java programming language which was developed at the University of Waikato, New Zealand. It has been used for data processing, classification, visualization, regression, clustering, and feature reduction and selection tasks [31]. Oversampling using different values of *K* in SMOTE, nearmiss, and Word2vec could also be considered. Nearmiss uses k-nearest neighbors to sample points from the majority classes. In Word2vec, each word is represented as a vector which also stores relational and semantic information between words. It can be used in place of TFIDF.

Bibliography

[1]. S. M. Vohra and J. B. Teraiya, A comparative study of sentiment analysis techniques, Journal of Information, Knowledge, and Research in Computer Engineering, vol. 2, no. 2, pp. 313–317, 2013.

[2]. R. Singh, R. Singh, and A. Bhatia, Sentiment analysis using machine learning technique to predict outbreaks and epidemics, International Journal of Advanced Science and Research, vol. 3, no. 2, pp. 19–24, 2018.

[3]. H. Wang, Z. Wang, and Y. Dong, Phase-adjusted estimation of the number of Coronavirus disease 2019 cases in Wuhan, China, Cell Discovery, vol. 6, art. no. 10, 2020.

[4]. M. S. Neethu and R. Rajasree, Sentiment analysis in Twitter using machine learning techniques, in Proceedings of the International Conference on Computing, Communications, and Networking Technologies, Tiruchengode, India, 2013.

[5]. M. Annett and G. Kondrak, A comparison of sentiment analysis techniques: Polarizing movie blogs, Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 5032, pp. 25–35, Springer, Berlin, 2008.

[6]. L. Yue, W. Chen, X. Li, and W. Zuo, A survey of sentiment analysis in social media, Knowledge and Information Systems, vol. 60, no. 2, pp. 617–663, 2019.

[7]. U. Naseem, I. Razzak, and M. Khushi, COVIDSenti: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis, IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 1003–1015, 2021.

[8]. C. Kaur and A. Sharma, Twitter sentiment analysis on Coronavirus using Textblob, EasyChair Preprint no. 2974, 2020.

[9]. R. P. Kaila and A. K. Prasad, Informational flow on Twitter–corona virus outbreak–topic modelling approach, International Journal of Advanced Research in Engineering and Technology, vol. 11, no. 3, pp. 128–134, 2020.

[10]. M. Alhajji, A. Al Khalifah, M. Aljubran, and M. Alkhalifah, Sentiment analysis of tweets in Saudi Arabia regarding governmental preventive measures to contain COVID-19, Preprint, 2020.

[11]. P. Tyagi, N. Goyal, and T. Gupta, Analysis of COVID-19 tweets during lockdown phases, in Proceedings of the International Conference on Information and Education Technology, Okayama, Japan, 2021, pp. 471–475.

[12]. R. Khan, P. Shrivastava, and A. Kapoor, Social media analysis with AI: sentiment analysis techniques for the analysis of Twitter COVID-19 data, Journal of Critical Reviews, vol. 7, no. 9, pp. 2761–2774, 2020.

[13]. A. Pak and P. Paroubek, Twitter as a corpus for sentiment analysis and opinion mining, in Proceedings of the International on Language Resources and Evaluation, Valletta, Malta, pp. 1320-1326, 2010.

[14]. G. Kalia, A research paper on social media: An innovative educational tool, Issues and Ideas in Education, vol. 1, no. 1, pp. 43–50, 2013.

[15]. S. Trinh, L. Nguyen, M. Vo, and P. Do, Lexicon-based sentiment analysis of Facebook comments in Vietnamese language, in Recent Developments in Intelligent Information and Database Systems, Studies in Computational Intelligence, Springer, Cham, Switzerland, vol. 642, pp. 263–276, 2016.

[16]. T. Carpenter and T. Way, Tracking sentiment analysis through Twitter, in Proceedings of the International Conference on Information and Knowledge Engineering, Las Vegas, NV, USA, 2012.

[17]. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, Lexicon-based methods for sentiment analysis, Computational Linguistics, vol. 37, no. 2, pp. 267–307, 2011.

[18]. A. Gelbukh, Natural language processing, in Proceedings of the International Conference on Hybrid Intelligent Systems, Rio de Janeiro, Brazil, 2005.

[19]. S. Joshi and D. Deshpande, Twitter sentiment analysis system, arXiv preprint arXiv: 1807.07752, 2018.

[20]. E. Kiely, L. Robertson, R. Rieder, D. A. Gore, Timeline of Trump's COVID-19 Comments, https://www.factcheck.org/2020/10/timeline-of-Trumps-covid-19-comments/, October 2020, [Online; accessed January 21, 2022].

[21]. M. Hagen, M. Potthast, M. Büchner, and B. Stein, Webis: An ensemble for Twitter sentiment detection, in Proceedings of the International Workshop on Semantic Evaluation, Denver, CO, USA, pp. 582–589, 2015.

[22]. J. Brownlee, Time series forecasting as supervised learning, https://machinelearningmastery.com/time-series-forecasting-supervised-learning/, 2016, [Online; accessed January 24, 2022].

[23]. A. P. Jain and P. Dandannavar, Application of machine learning techniques to sentiment analysis, in Proceedings of the International Conference on Applied and Theoretical Computing and Communication Technology, Bangalore, India, pp. 628–632, 2016.

[24]. W. Medhat, A. Hassan, and H. Korashy, Sentiment analysis algorithms and applications: A survey, Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113, 2014.

[25]. F. Pérez and B. E. Granger, IPython: A system for interactive scientific computing, Computing in Science & Engineering, vol. 9, no. 3, pp. 21–29, 2007.

[26]. T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, and P. Ivanov, Jupyter Notebooks-a publishing format for reproducible computational workflows, in Positioning and Power in Academic Publishing: Players, Agents, and Agendas, Proceedings of the International Conference on Electronic Publishing, Göttingen, Germany, pp. 87-90, 2016.

[27]. N. Bell, L. N. Olson, and J. Schroder, PyAMG: Algebraic multigrid solvers in Python, Journal of Open Source Software, vol. 7, no. 72, art. no. 4142, 2022.

[28]. M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos, Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches, IEEE Computational Intelligence Magazine, vol. 13, no. 4, pp. 59–76, 2018.

[29]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W.P. Kegelmeyer, Smote: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[30]. J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, Evaluating the quality of machine learning explanations: A survey on methods and metrics, Electronics, vol. 10, no. 5, pp. 593, 2021.

[31]. I. H. Witten, E. Frank, A. Mark, and J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, Chennai, India, 2016.